IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ignasi Casamayor
2024/September

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
  - Data collection API with Webscraping
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - Exploratory Data Analysis for Data Visualization
  - Interactive Visual Analytics and Dashboard
  - Predictive Analysis
- Summary of all results
  - Exploratory Results
  - Predictive Analysis

Introduction

- Project background and context

Space X says Falcon 9 rocke launches with a cost of only 62 million dollars. Other providers' costs arrive to 165 million. It looks like much of the savigs are due to reuse of first stage. Hence, if it is possible to determine if the first stage will land, then we can determine the launching cost. This information can be used if an alternate company wants to bid against space X. The goal of the project is to create a ML pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

  - What factors determine if a rocket will land with success?
  - The successful landing can be determined bythe interaction among features?
  - What conditions can ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected using SpaceX API and Webscraping from Wikipedia
- Perform data wrangling
  - One-hot encoding was applied
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data collection was done through get_requests to the SpaceX API.
- .json() function was called to normalized the decoded responses into a pandas df.
- Afterwards, a cleansing data was done.
- Additionaly, a web scraping from Wikipedia was done in order to get the Falcon 9 launch records using BeautifulSoup.
- The goals were: extract the data, parse the HTML table and convert the data into padas df for future analysis.

Data Collection – SpaceX API

- Collection of data was done using the GET request.
- Some cleansing and basic normalization was done.

- The link to the file: https://github.com/IgnasiCSO/IBM-Final/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Webscrapping to Falcon 9 launch records with BeautifulSoup were done.
- The table was parsed and converted into a pandas dataframe.
- The link: https://github.com/IgnasiCSO/IBM-Final/blob/main/jupyter-labs-webscraping.ipynb

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_H
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP respons

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response= requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup= BeautifulSoup(response, 'html.parser')
```

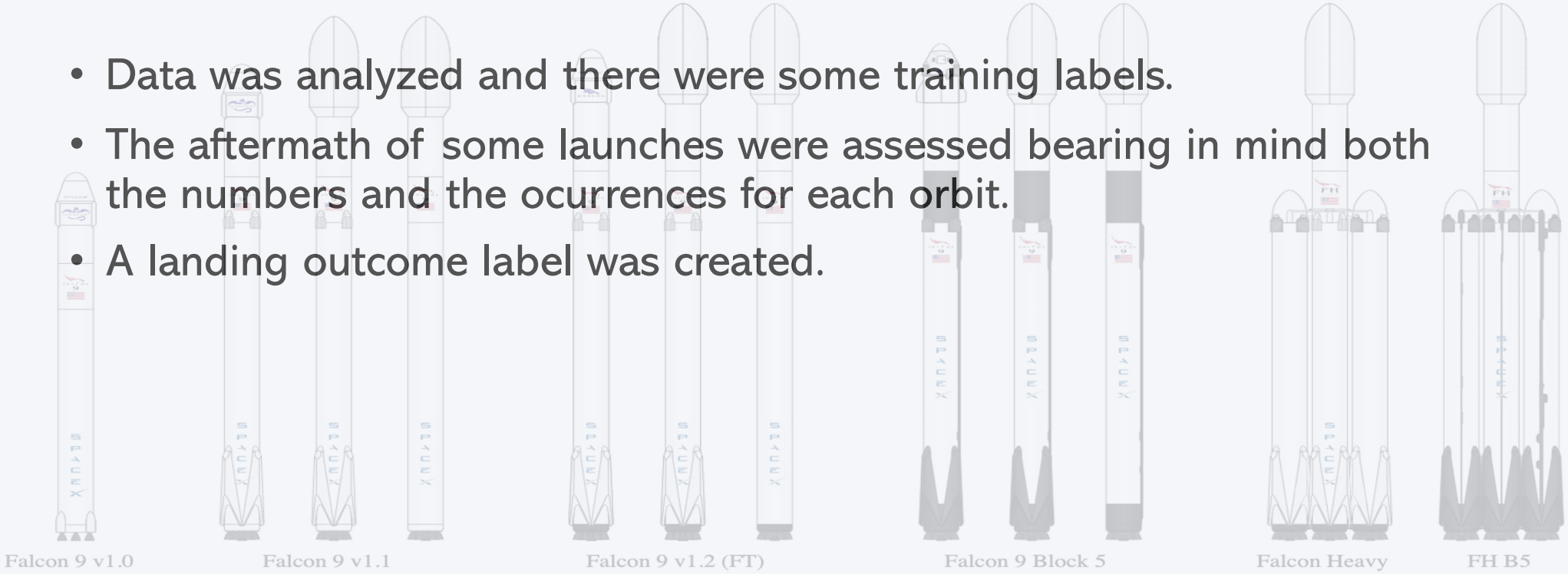
Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

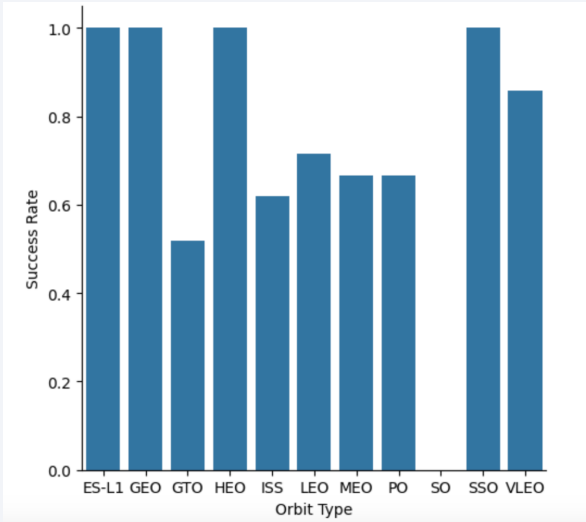
- Data was analyzed and there were some training labels.
- The aftermath of some launches were assessed bearing in mind both the numbers and the ocurrences for each orbit.
- A landing outcome label was created.

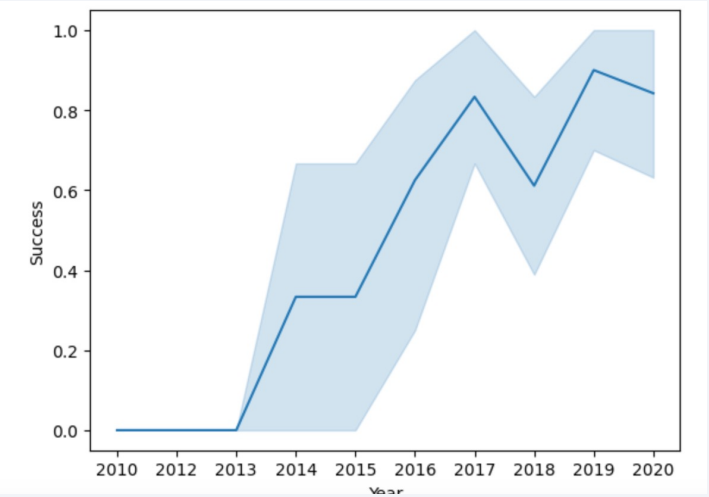


- More to see on: https://github.com/IgnasiCSO/IBM-Final/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- There was an exploration using DataViz in order to see possible relationships among variables.
- Here, it is possible to understand 2 take aways:
  - The orbits have a key relevance in the launching success. (Picture above).
  - The launching success is doing better through the years. (Picture below).
  - Link to a more exhaustive information: https://github.com/IgnasiCSO/IBM-Final/blob/main/edadataviz.ipynb





EDA with SQL

- SpaceX dataset was loaded into sql.
- EDA was applied with SQL to assess the data. Some queries were done to better understand the environment:
  - Names of the launch sites.
  - Palyload mass carried (both total and average).
  - Results of missions.
  - Failed landing outcomes in drone ships.

- Link to info: https://github.com/IgnasiCSO/IBM-Final/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Launch site were mark. Some objects were map-added as markers too.
- Failure or success launch outcome where assingte to classes 0 and 1.
- Clusters were created.
- Distances between sites and items from their vicinities were measured.

- Link: https://github.com/IgnasiCSO/IBM-Final/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- An interactive dashboard with Plotly dash was created.
- Pie charts were created.
- Plotted scatters graphs showing relationships were created as well.

- Link: https://github.com/IgnasiCSO/IBM-Final/blob/main/spacex_dash_app.py

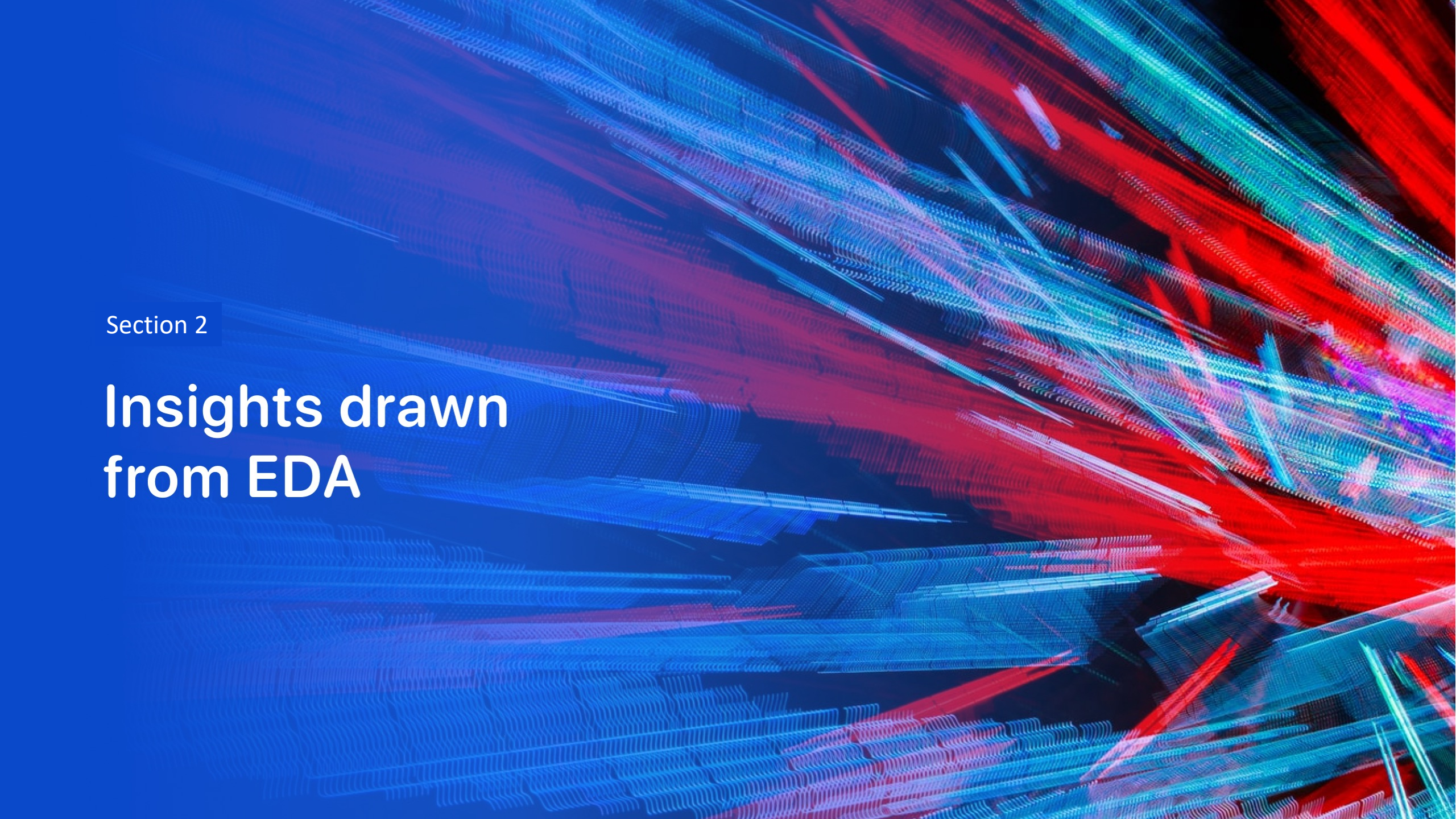
Predictive Analysis (Classification)

- Data was loaded using numpy and pandas. Data was transform and split into training and test.
- Different ML models are created and different hiperparameters were tune using GridSearchCV.
- The metric accuracy was used to understand the better approach.

- Link: https://github.com/IgnasiCSO/IBM-Final/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

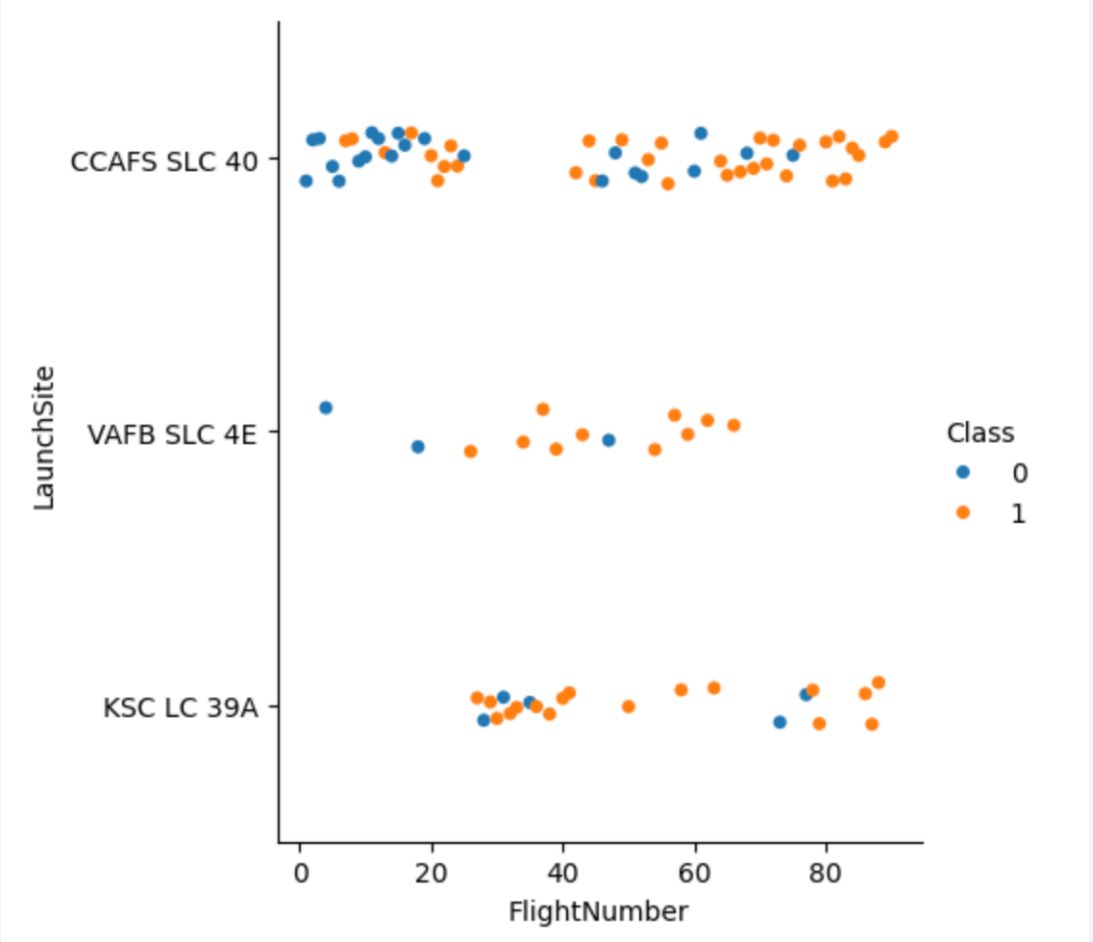
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

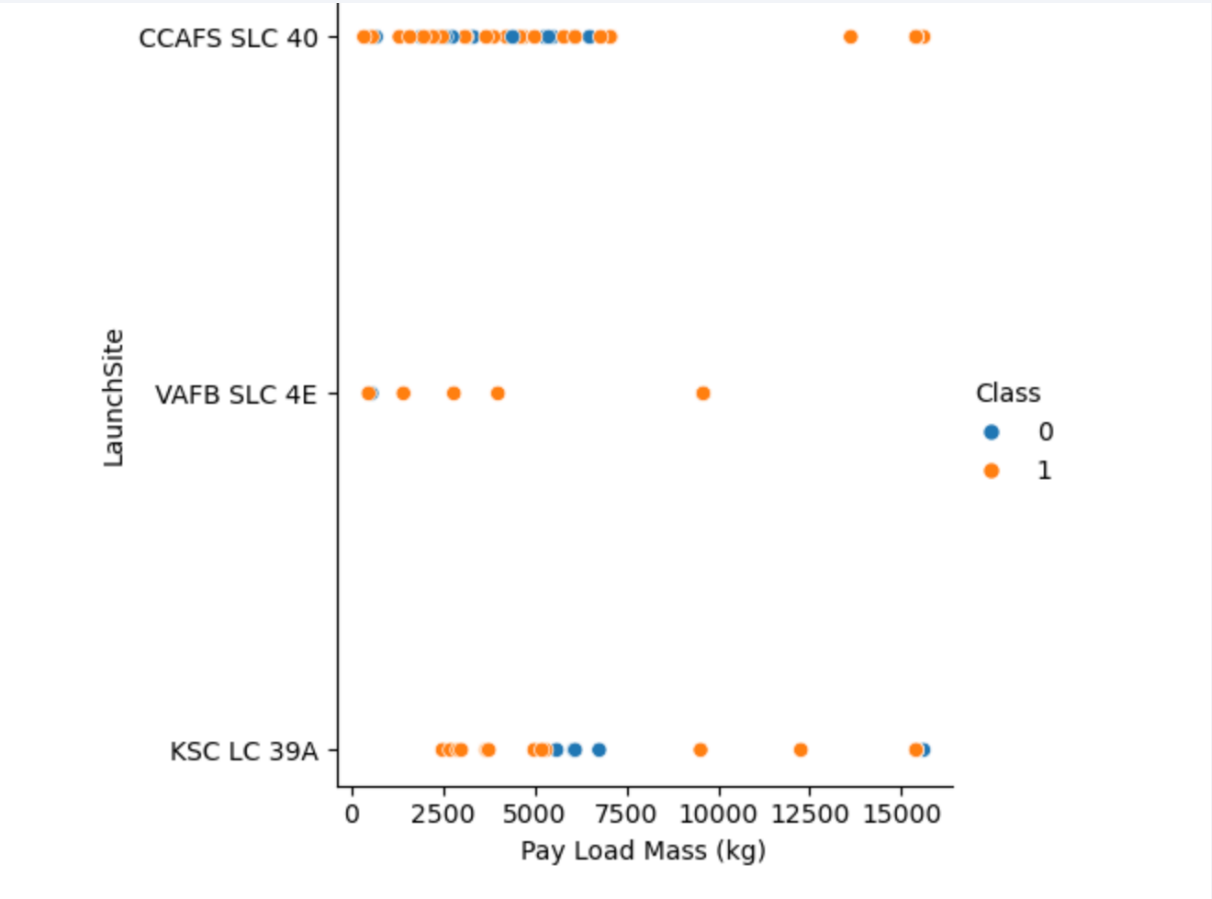
Flight Number vs. Launch Site

- From the plot it is possible to learn that:
  - As there number of flights increase, the success launches do better too.
  - The conclusion is independent from the launch site.



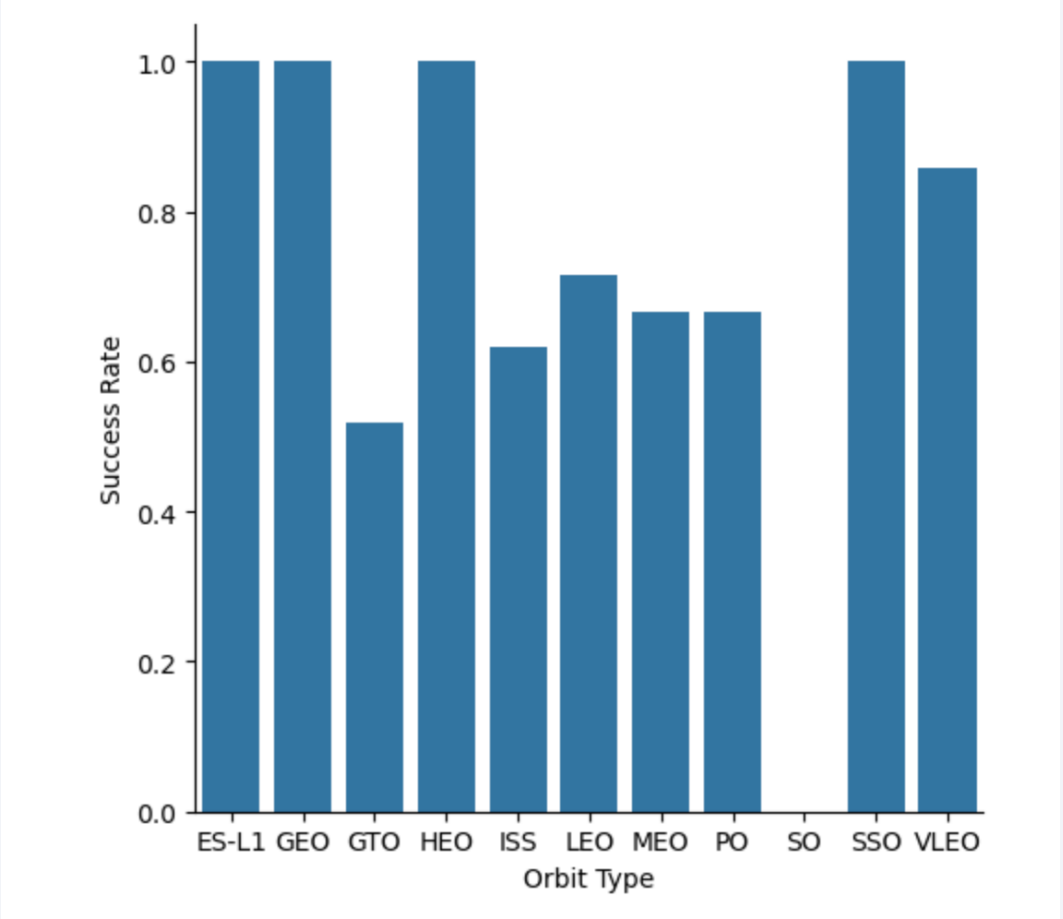
Payload vs. Launch Site

- The higher the payload mass, the higher the rocket's success rate.



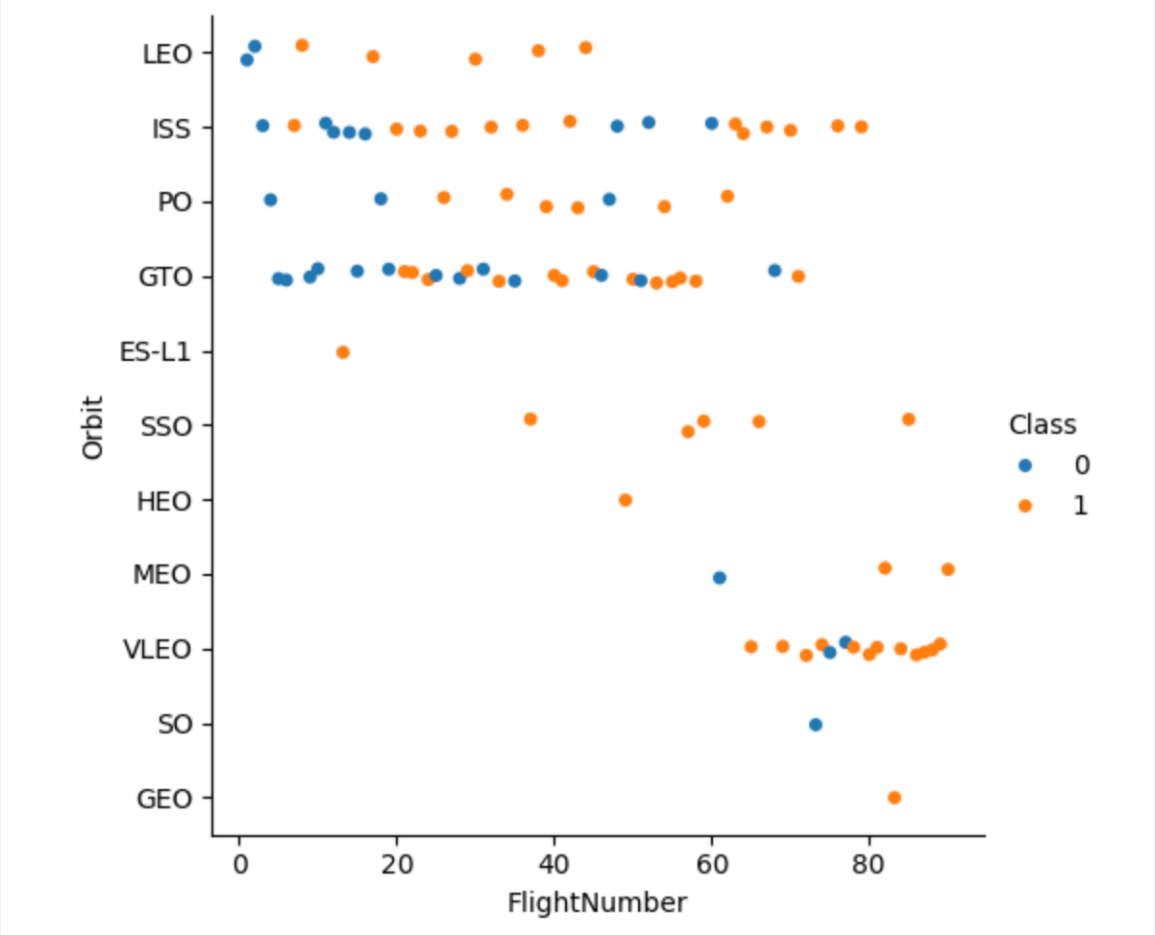
Success Rate vs. Orbit Type

- Most successful orbits are:
  - ES-L1
  - GEO
  - HEO
  - SSO
- The worst:
  - SO



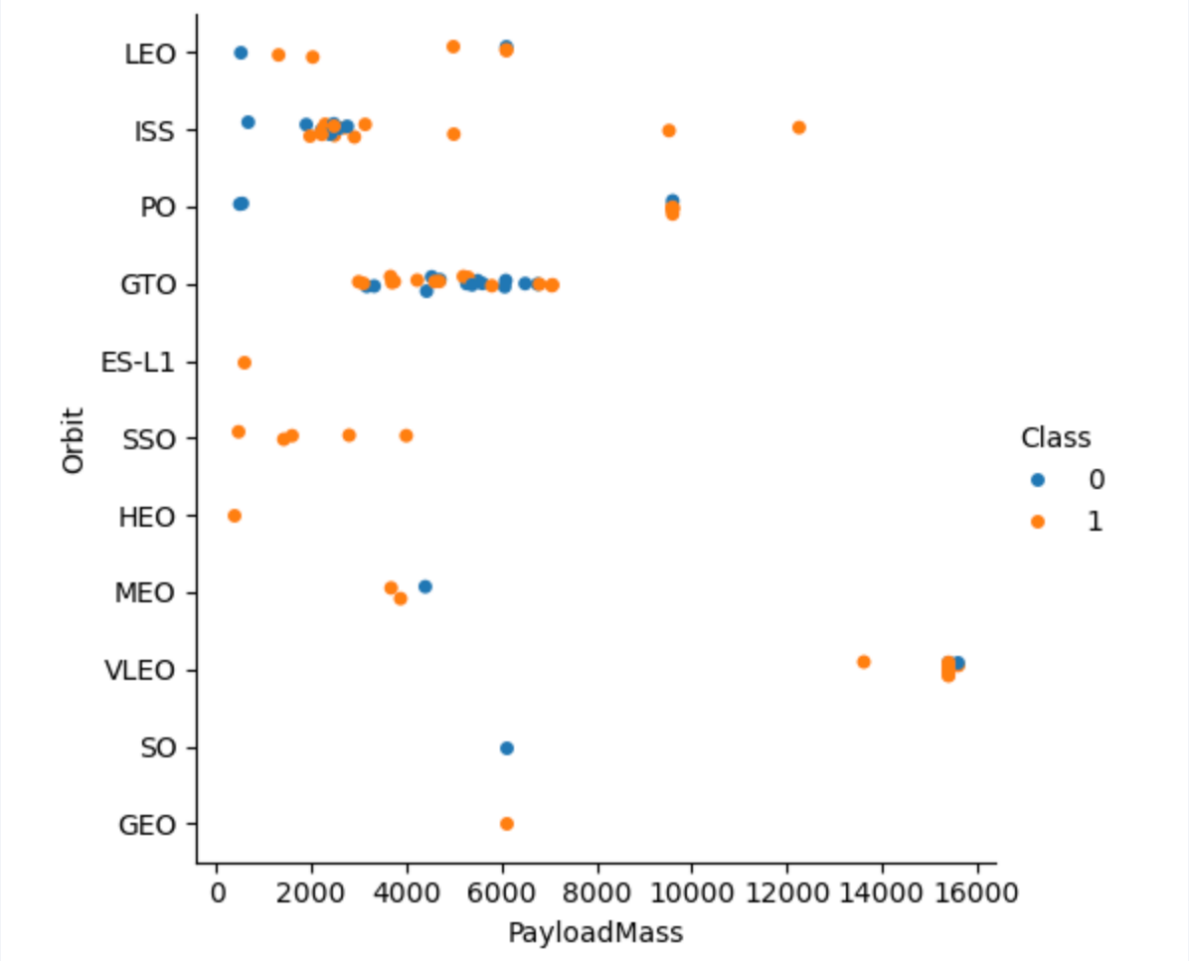
Flight Number vs. Orbit Type

- Generally speaking there is no correlation between orbits and flight numbers to determine if a launch will succeed or fail.



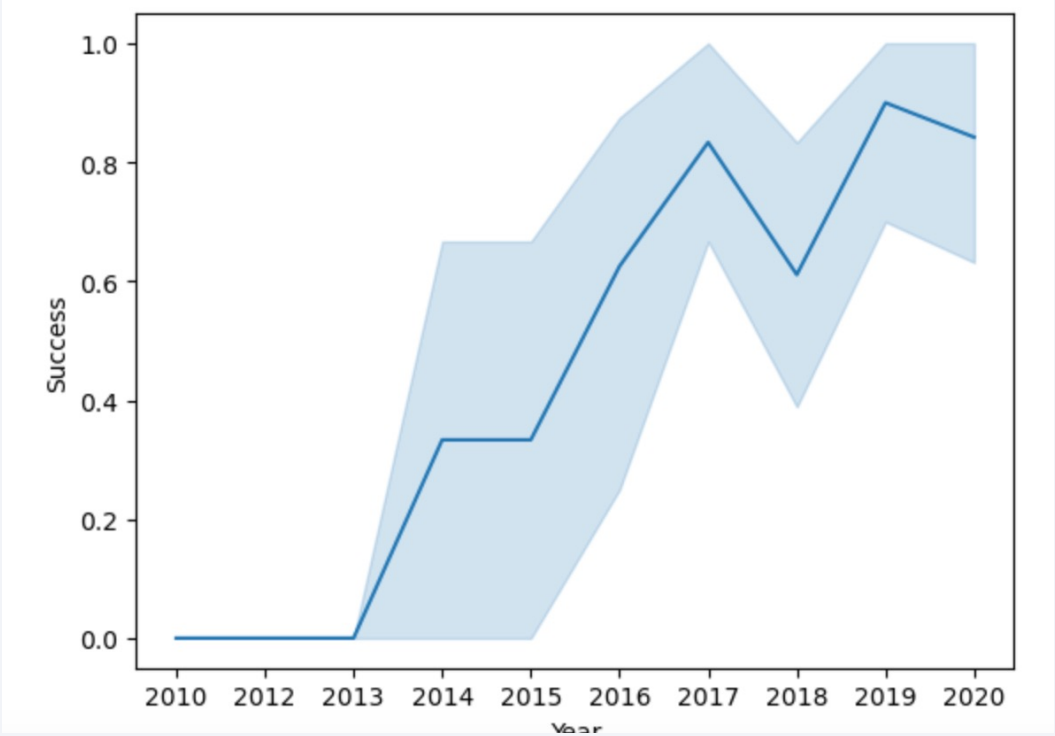
Payload vs. Orbit Type

- Orbits usually perform better with higher payload masses (as ISS, VLEO or PO).
- Other orbits are independent to the payload mass (as SSO, ES-L1 or HEO).



Launch Success Yearly Trend

- Since 2013, every year the success rate is doing better (with just a bump in 2018).



All Launch Site Names

- The keyword DISTINCT was used to learn the unique launch sites.

Display the names of the unique launch sites in the space mission

In [23]:
```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;
```

```
 * sqlite:///my_data1.db
Done.
```

Out[23]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- Following instruction shows the launche site names that begin with ‘CCA’

Display 5 records where launch sites begin with the string 'CCA'

In [24]:
```
%sql SELECT LAUNCH_SITE FROM SPACEXTBL WHERE (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Out[24]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- The result of the total payload carried by boosters as follows below:

Display the total payload mass carried by boosters launched by NASA (CRS)

In [40]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER= 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

Out[40]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Here the result of the average payload mass carried just by the booster version F9 v1.1.

Display average payload mass carried by booster version F9 v1.1

In [39]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION= 'F9 v1.1';
```

```
 * sqlite:///my_data1.db
Done.
```

Out[39]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Couldn't find the proper instruction to find out the first successful ground landing date.

```
In [47]: %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE (LANDING__OUTCOME) LIKE 'TRUE';
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: SELECT MIN(DATE) FROM SPACEXTBL WHERE (LANDING__OUTCOME) LIKE 'TRUE' ;]
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Some filtering was used to see the successful drone ship landing with payload between 4000 and 6000.

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [61]:
```
%sql SELECT PAYLOAD FROM SPACEXTBL WHERE LANDING_OUTCOME= 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 40
```

```
 * sqlite:///my_data1.db
Done.
```

Out[61]:

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- Here you have total number of successful and failure mission outcomes.

```
In [48]: %sql SELECT MISSION_OUTCOME, COUNT(*) AS TOTAL_NUMBER FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
 * sqlite:///my_data1.db
Done.
```

Out[48]:

Mission_Outcome	TOTAL_NUMBER
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Subqueries are used here to determined the boosters that carried the maximum payload.

```
In [49]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTE
 * sqlite:///my_data1.db
Done.
```

Out[49]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- Some detailed info was searched using a more detailed instruccions as it can be seen here.

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [81]:
```
%sql SELECT SUBSTR(DATE,6,2) AS MONTH, DATE, BOOSTER_VERSION, LAUNCH_SITE, LANDING_OUTCOME FROM SPACEXTBL WHERE (
```

```
* sqlite:///my_data1.db
Done.
```

Out[81]:

MONTH	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

- Some grouping and ordering was applied to get the proper data

Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

In [76]:
```
%sql SELECT(LANDING_OUTCOME), COUNT(*) AS COUNT_OUTCOMES FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017
```

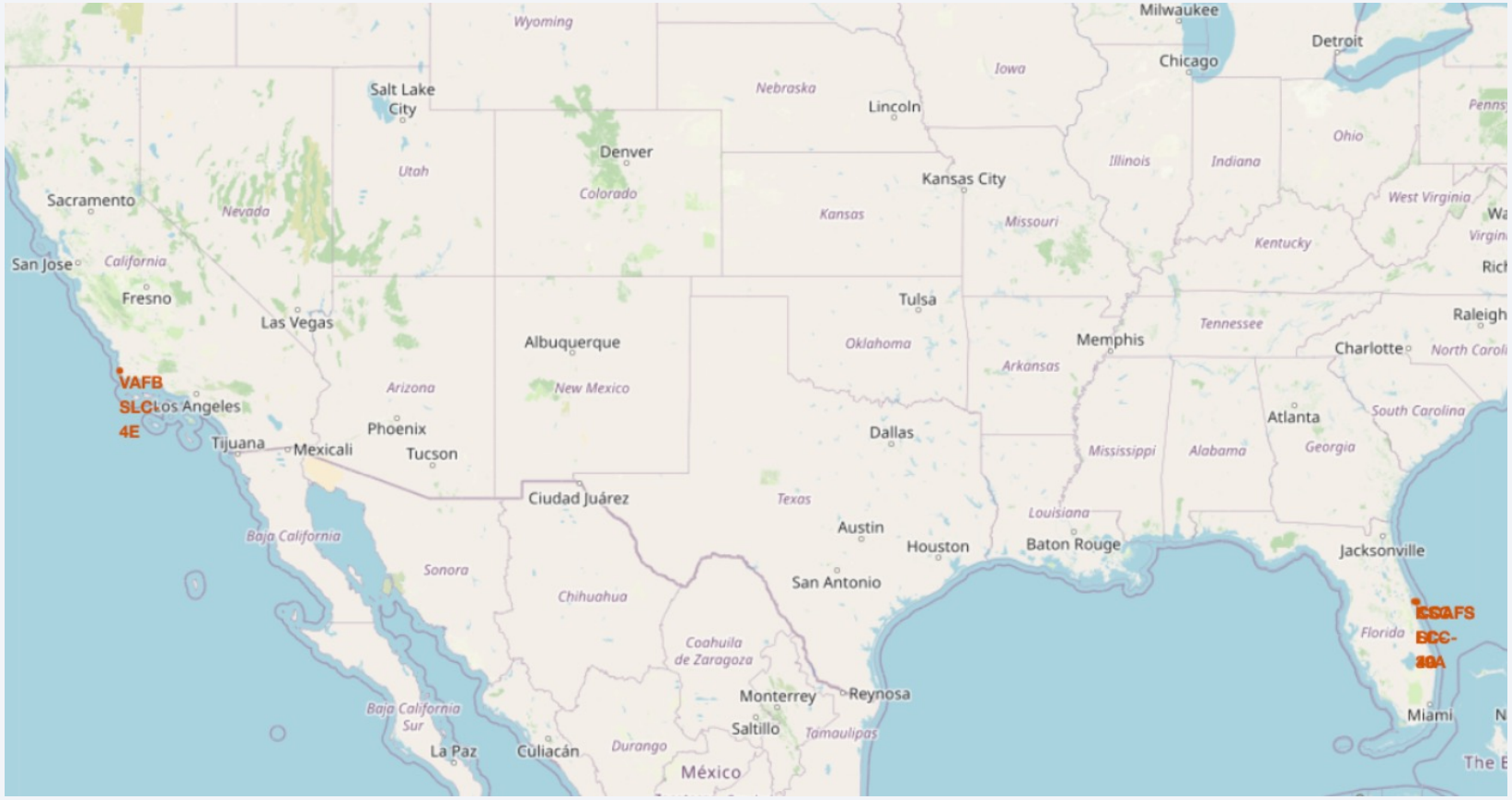
```
* sqlite:///my_data1.db
Done.
```

Out[76]:

Landing_Outcome	COUNT_OUTCOMES
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1


Section 3
Launch Sites
Proximities Analysis

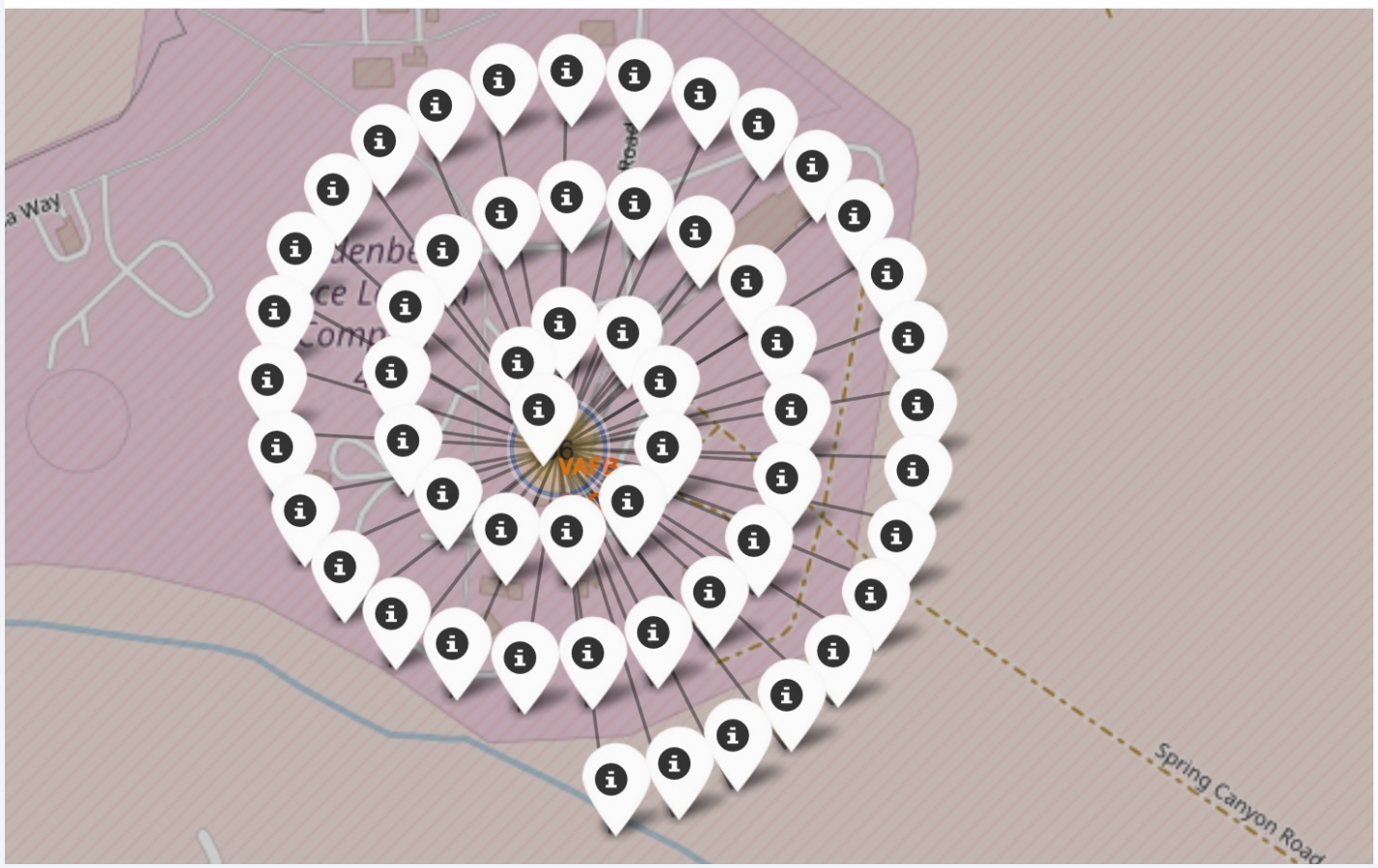
Launch sites



SpaceX launch sites are settled at both sides of the US

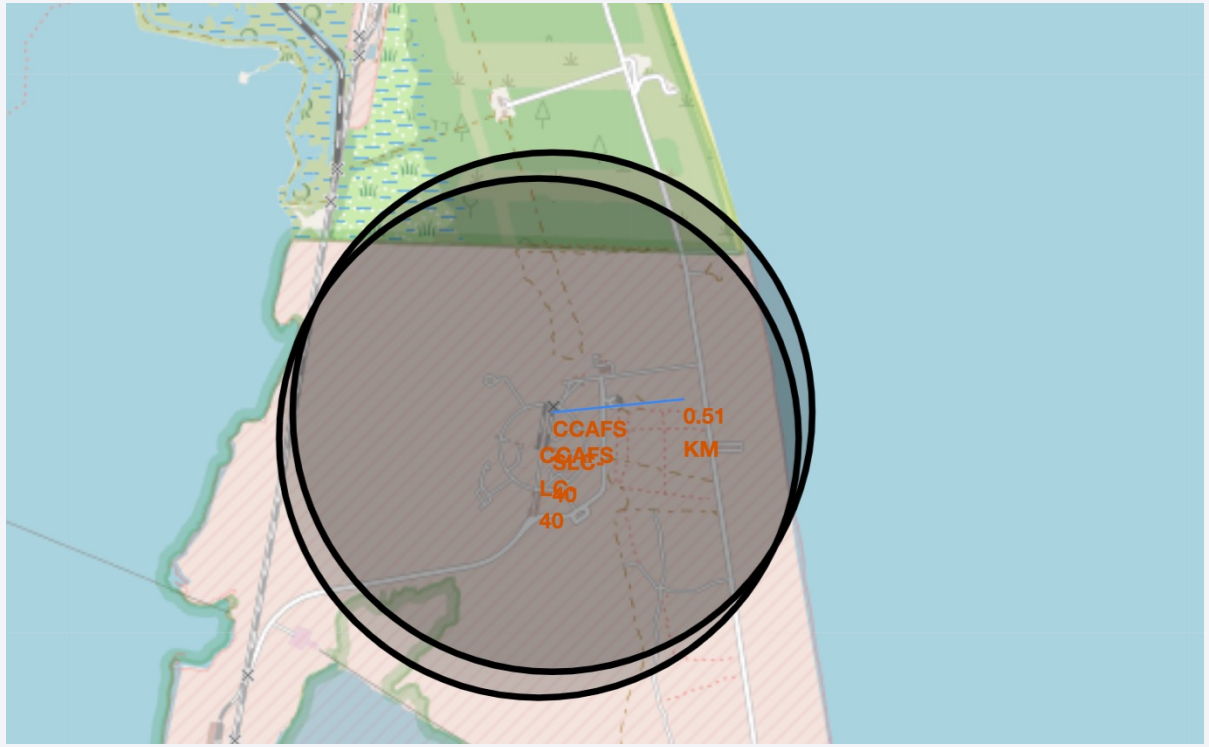
Markers showing launch sites

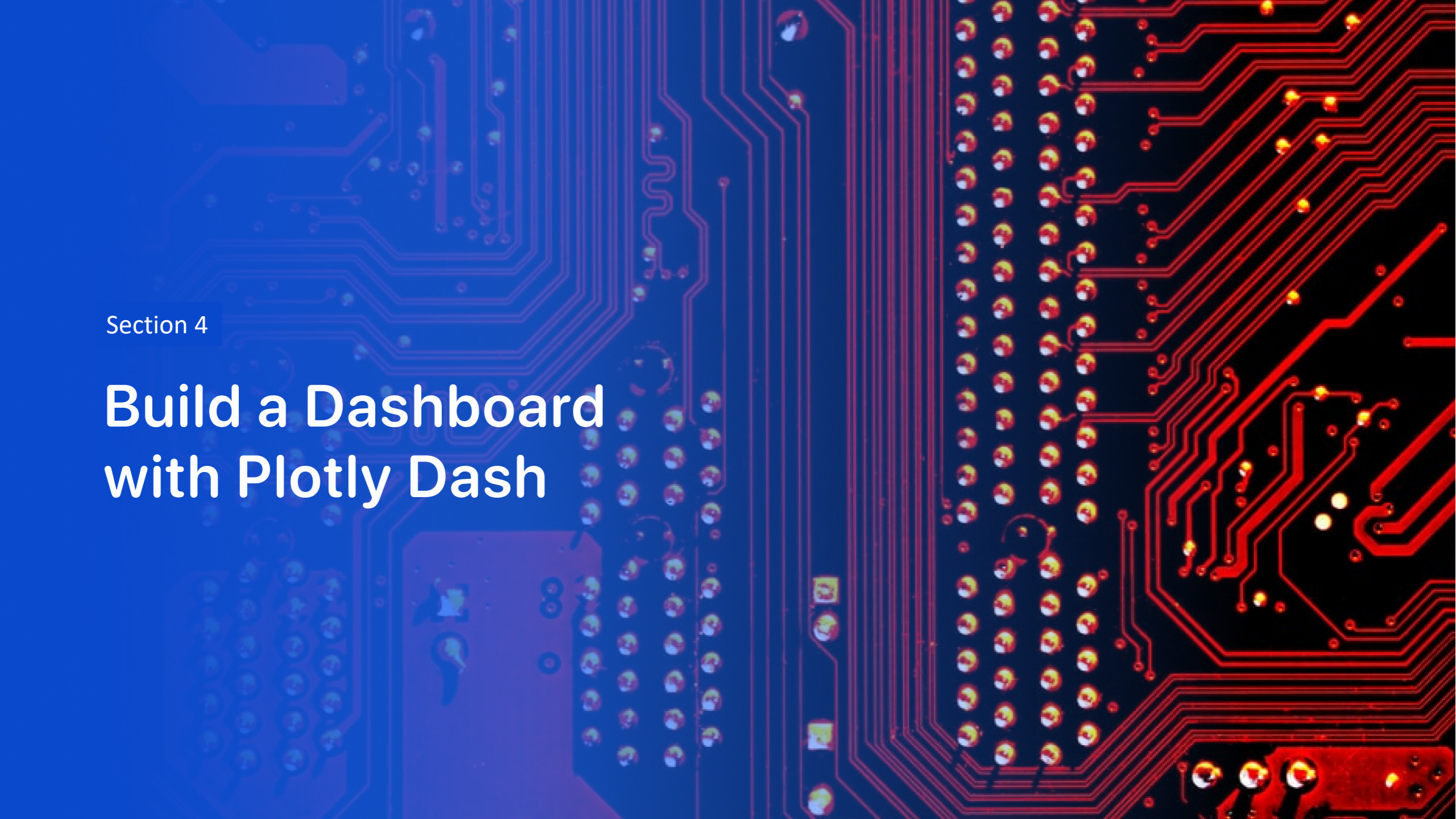
In following picture it is possible to see all the launches from the VAFB SLC-4E station



Distances to landmarks from launch sites

- There are no special landmarks that could disrupt regular launching.
- Here you have an example of a rouad from CCAFS station... And the coast is even further!




Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The Decision Tree model is the best algorithm with the greatest accuracy.

Find the method performs best:

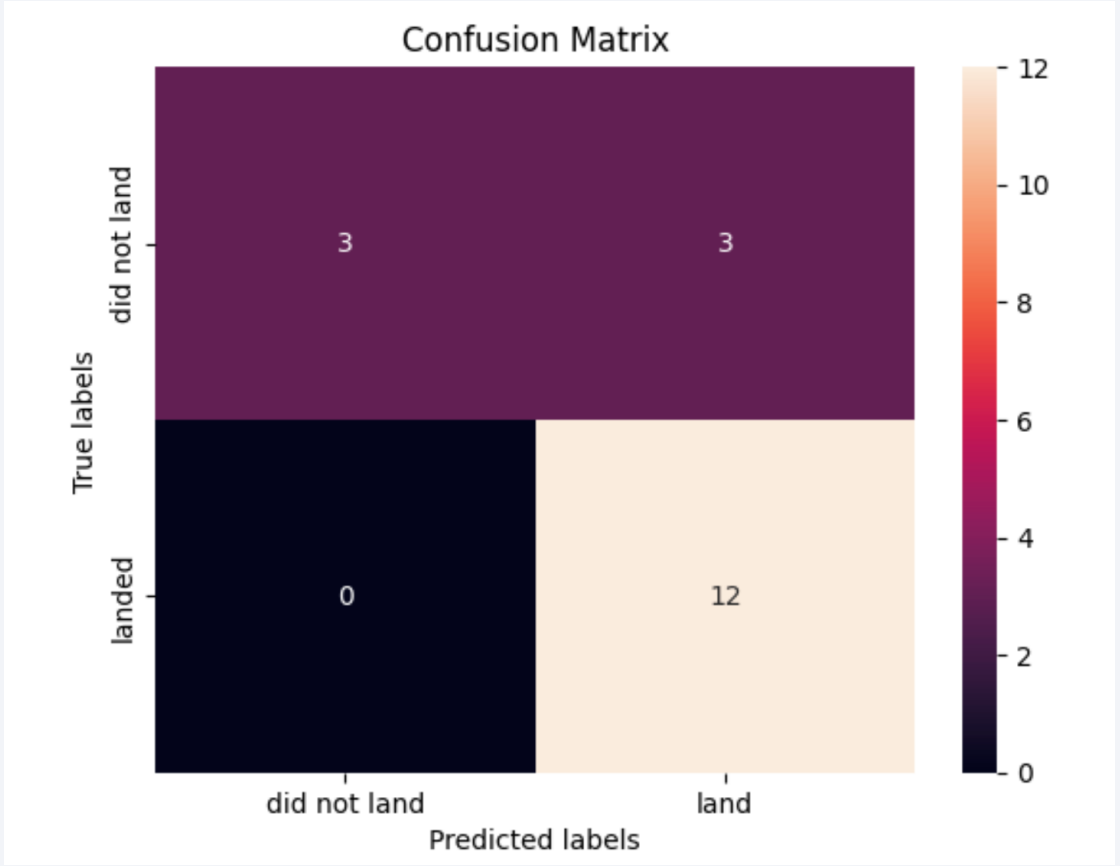
```
[39]: models={'KNeighbors': knn_cv.best_score_, 'DecisionTree': tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

      bestalgorithm= max(models, key=models.get)
      print('Best model ', bestalgorithm, 'with a score of', models[bestalgorithm])
```

```
Best model  DecisionTree with a score of 0.8625
```

Confusion Matrix

- Confusion matrix show that the classifier can distinguish among classes.
- The problem here are the false positives.



Conclusions

- The more flight we have at a launch site, the greater the success rate.
- Launch success rate is doing better through time.
- Some orbits perform better than others (ES-L1, GEO, HEO, SSO).
- KSC LC-39A has the better results of success.
- Decision trees are the best algorithm to use.

Thank you!